Micro-Processor and Embedded Systems Lab-Session 2 Report for Instruction Memory and Data Memory

First Name : Yagna Srinivasa Harsha

Last Name : Annadata

Net ID : yxa210024

UTD ID : 2021641648

Email Id : yxa210024@utdallas.edu

Date : 2 September 2022.

First Name: Leela Sumanth

Last Name: Narla

Net ID: lxn220007

UTD ID: 2021672975

Email Id: lxn220007@utdallas.edu

Date: 2 September 2022

AIM: This week we were asked to create an Instruction and Data Memory modules. I was able to successfully create them.

Summary:

1. Using Nomachine access the VIVADO By sourcing **/proj/cad/startup/profile.xilinx_vivado_18.3.**
2. Command to open the tool is **vivado&**
3. Create a new project on the software for ALU and registers.
4. ALU and register file Verilog codes along with the test benches have been complied successfully.
5. Ran the behavioral simulation for the test bench codes.

Instruction Memory

The Instruction Memory (IM) stores all the prefetch instructions. It is composed of 5 major components: the PC (Program Counter) Unit, PC Decoder, INBUF (an input buffer to the IM), IM Storage, OUTBUF (an output buffer to the internal bus)

The output from the PC Decoder asserts one of the memory locations' READ or WRITE signal. When the WRITE signal of an IM Memory location is asserted, the value in the INBUF is written into that memory location. On the other hand, when the READ signal of an IM Memory location is asserted, the value from that storage location is read to the OUTBUF

Data Memory

The Data Memory (DM) stores all the variables.

The output from the PC Decoder asserts one of the memory locations' READ or WRITE signal. When the WRITE signal of an DM Memory location is asserted, the variable or the data in the INBUF is written into that memory location. On the other hand, when the READ signal of an IM Memory location is asserted, the data or variable from that storage location is read to the OUTBUF.

Data bytes are arbitrary bits of information to be processed by the computer. Instructions are special sequences of bytes that cause the microprocessor to perform specific tasks.

Conclusion:

We were able to successfully create an Instruction and Data Memory. I would like to implement all the modules created into an MCU

Attachments:

We have uploaded the codes and testbenches for an Instruction and Data Memory in the attachments along with a screenshot of the simulation.